## Amendments to the Claims

This listing of the claims will replace all prior versions, and listings of the claims in this application.

### Listing of Claims

1. (Currently amended) A system for secure data transfer over a network, the system comprising:

memory;

a memory controller configured to transfer data received from the network to the memory;

a network interface coupled to the memory controller, the network interface comprising:

a first data moving unit (DMU) configured to exchange secure data with a first portion of the network;

a second DMU configured to exchange non-secure data with a second portion of the network; and

a processor coupled to the memory controller, the processor including:

logic configured to retrieve a portion of the data from the memory using the memory controller;

logic configured to perform security operations on the retrieved portion of the data; and

logic configured to store the operated-on portion of the data in the memory using the memory controller;

wherein the memory controller is further configured to transfer the operated-on portion of the data from the memory to the network.

2. (Currently amended) The system of claim 1,~~ comprising a network interface coupled to memory controller, the network interface comprising:~~ wherein the [[a]] first and second DMUs directly ~~data moving unit (DMU)~~ communicate with the

~~configured to exchange secure data with a~~ first and seconds portions ~~portion~~ of the network~~; and a second DMU configured to exchange non-secure data with a second portion of the network~~.

3. (Original) The system of claim 2, wherein the network interface comprises:

a first serializer/deserializer (SERDES) circuit coupled between the first DMU and the first network portion and a second SERDES coupled between the second DMU and the second network portion, each SERDES configured to convert serial data received from the respective network portions to a parallel format and to convert parallel data received from the respective DMUs to a serial format.

4. (Original) The system of claim 1, wherein the logic configured to perform security operations comprises:

logic configured to obscure the portion of the data when the retrieved portion is non-secure data;

logic configured to decipher the portion of the data when the retrieved portion is secure data; and

logic configured to determine an integrity of the portion of data.

5. (Original) The system of claim 1, wherein the processor comprises:

logic configured to perform quality-of-service (QoS) operations on the data in coordination with performing the security operations.

6. (Currently amended) The system of claim 5, wherein the logic configured to perform QoS operations comprises:

logic configured to identify an information flow associated with the portion of the data;

logic configured to determine a priority of the information flow; and

logic configured to schedule at least one of the retrieving the portion of the data and the transferring of the operated-on portion of the data from memory based on the priority of the information flow associated with the portion of the data.

7. (Original) The system of claim 6, wherein the processor comprises:

logic configured to decipher the portion of the data prior to the identifying of the information flow when the retrieved portion is secure data; and

logic configured to obscure the portion of the data after the identifying of the information flow when the retrieved portion is non-secure data.

8. (Original) The system of claim 1, wherein the processor comprises:

logic configured to compress the portion of the data using the processor prior to performing the security operations when the retrieved portion is non-secure data; and

logic configured to decompress the portion of the data in the processor after performing the security operations when the retrieved portion is secure data.

9. (Original) The system of claim 1, wherein the memory includes a memory block having a plurality of memory banks, the memory controller comprising:

logic configured to reference the plurality of memory banks in a sequence that minimizes a memory access time.

10. (Original) The system of claim 1, wherein the memory controller comprises:

logic configured to include a request to reference the memory into one of a group of read requests and a group of write requests; and

logic configured to execute all requests included in one of the groups of read requests and write requests before executing a request included in the other group.

11. (Original) The system of claim 10, comprising:

logic configured to include error correction code with the data transferred to or stored in the memory; and

logic configured to detect and correct errors in the data retrieved or transferred from the memory based on the error correction code included with the data.

12. (Currently amended) A method for secure data transfer over a network, the method comprising:

transferring data from the network to memory using a memory controller;

retrieving a portion of the data from the memory into a processor using the memory controller, wherein portions of the data having higher priority information flow are retrieved before portions of the data having lower priority information flow;

performing security operations on the retrieved portion of the data using the processor;

storing the operated-on portion of the data in the memory using the memory controller; and

transferring the operated-on portion of the data from the memory to the network using the memory controller.

13. (Original) The method of claim 12, wherein the security operations comprise at least one of:

obscuring the portion of the data when the retrieved portion is non-secure data;

deciphering the portion of the data when the retrieved portion is secure data; and

determining an integrity of the portion of data.

14. (Original) The method of claim 12, comprising:

performing quality-of-service (QoS) operations on the data in coordination with performing the security operations using the processor.

15. (Original) The method of claim 14, wherein the QoS operations comprise:

identifying an information flow associated with the portion of the data;

determining a priority of the information flow; and

scheduling at least one of the retrieving the portion of the data and the transferring the operated-on portion of the data from memory based on the priority of the information flow associated with the portion of the data.

16. (Original) The method of claim 15, comprising:

deciphering the portion of the data prior to the identifying of the information flow when the retrieved portion is secure data; and

obscuring the portion of the data after the identifying of the information flow when the retrieved portion is non-secure data.

17. (Original) The method of claim 12, comprising:

compressing the portion of the data using the processor prior to performing the security operations when the retrieved portion is non-secure data; and

decompressing the portion of the data in the processor after performing the security operations when the retrieved portion is secure data.

18. (Original) The method of claim 12, comprising:

including a request to reference the memory into one of a group of read requests and a group of write requests; and

executing all requests included in one of the groups of read requests and write requests before executing a request included in the other group.

19. (Original) The method of claim 18, wherein the executing all requests included in one of the groups of read requests and write requests occurs when a sum of the requests included in one of the groups corresponds to a predetermined amount of the memory.

20. (Original) The method of claim 12, comprising:

including error correction code with the data transferred to or stored in the memory; and

at least one of detecting and correcting errors in the data retrieved or transferred from the memory based on the error correction code included with the data.

21. (Original) The method of claim 12, comprising:

referencing portions of the memory in a sequence that minimizes a memory access time.

22. (Currently amended) A computer readable medium containing a computer program for secure data transfer over a network, wherein the computer program comprises executable instructions for:

transferring data from the network to memory using a memory controller;

retrieving a portion of the data from the memory into a processor using the memory controller;

performing security operations on the retrieved portion of the data using the processor;

storing the operated-on portion of the data in the memory using the memory controller; and

transferring the operated-on portion of the data from the memory to the network using the memory controller, wherein operated-on portions of the data having higher priority information flow are transferred before portions of the data having lower priority information flow.

23. (Original) The computer readable medium of claim 22, wherein the computer program comprises executable instructions for:

obscuring the portion of the data when the retrieved portion is non-secure data;

deciphering the portion of the data when the retrieved portion is secure data; and

determining an integrity of the portion of data.

24. (Original) The computer readable medium of claim 22, wherein the computer program comprises executable instructions for:

performing quality-of-service (QoS) operations on the data in coordination with performing the security operations using the processor.

25. (Original) The computer readable medium of claim 24, wherein the computer program comprises executable instructions for:

identifying an information flow associated with the portion of the data;

determining a priority of the information flow; and

scheduling at least one of the retrieving the portion of the data and the transferring the operated-on portion of the data from memory based on the priority of the information flow associated with the portion of the data.

26. (Original) The computer readable medium of claim 22, wherein the computer program comprises executable instructions for:

compressing the portion of the data using the processor prior to performing the security operations when the retrieved portion is non-secure data; and

decompressing the portion of the data in the processor after performing the security operations when the retrieved portion is secure data.